

101192-656

1N-81-CR

70537

p-31

Comparison of Display Enhancement with Intelligent Decision-Aiding

Proposed Research
and
Preliminary Analyses

by
Alex Kirlik
Wendy J. Markert
Merrick Kossack

Center for Human-Machine Systems Research
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332
(404) 894-2373

February 3, 1992

ABSTRACT

Currently, two main approaches exist for improving the human-machine interface component of a system in order to improve overall system performance—display enhancement and intelligent decision-aiding. Each of these two approaches has its own set of advantages and disadvantages, as well as introduce its own set of additional performance problems. These characteristics should help identify which types of problem situations and domains are better aided by which type of strategy. This report first describes the characteristic issues of these two decision-aiding strategies. Then differences in expert and novice decision-making are described in order to help determine whether a particular strategy may be better for a particular type of user. Finally, research is outlined to compare and contrast the two technologies, as well as to examine the interaction effects introduced by the different skill levels and the different methods for training operators.

(NASA-CR-189895) COMPARISON OF DISPLAY
ENHANCEMENT WITH INTELLIGENT DECISION-AIDING
(Georgia Inst. of Tech.) 31 p

N92-25726

Unclas
G3/81 0070537

Introduction

Currently, two main approaches exist for improving the human-machine interface component of a system in order to improve overall system performance—display enhancement and intelligent decision-aiding. These two approaches stem from the two main ways in which to aid human performance. People can be aided in what they perceive (by making important information more easily identified) as is the chief concern of display enhancement, or they can be aided in what they do with the information they perceive (making it easier to perform operations, etc.) which is a goal of intelligent decision-aiding.

These two technologies have their own sets of advantages and disadvantages, as well as introduce their own sets of error modes (or new performance problems). Each approach has characteristics which should help determine which problem situations are better handled by one than the other, as well as what type of user (in terms of amount of experience) is better aided by which type of human-machine interaction.

This research intends to compare and contrast these two technologies—display enhancement and intelligent decision-aiding—to determine which types of human-machine systems are better improved by each. Specifically, the domain examined here is a complex, dynamic decision-making task, but it is hoped that a thorough study will lead to conclusions which can be made about other domains as well. A look will also be taken to determine if different skill levels of users are aided in different ways by these two approaches. Also, the possible error modes introduced by each type of aiding will be examined.

This report is divided into five main sections. The first section will examine the characteristic issues of intelligent decision aids, as well as determine the possible error modes introduced by this type of aiding. The next section will discuss the issues surrounding the technology of display enhancements, as well as determine the possible error modes introduced by this technology. The third section will identify some of the differences in expert and novice

decision-making which may help identify how different types of aiding systems may aid different skill levels in different ways. Then research will be outlined in which to compare and contrast the technologies of intelligent decision-aiding with display enhancements, as well as the effects found when using both with different skill levels of operators. Finally, the expected contributions of this research will be described.

Characteristic Issues of Intelligent Decision-Aiding

Intelligent decision-aiding is an area which has been the focus of much research in the hopes of discovering ways to improve the performance of human-machine systems. Woods [1986], Woods and Roth [1988], Norman [1988], and Hopkin [1988] are just a few of the many researchers who have tried to determine what makes a good intelligent decision aid.

Among the issues which need to be addressed in the design of such a system is the role of the decision aid in the system. Woods and Roth [1988] discuss the role of the aid as an instrument (as opposed to as a prosthesis which is meant to replace people or to fill a human deficiency). As an instrument the decision aid is to be used as a reference or extra source of information for the human problem solver. This perspective leaves the human in control and performing more than mere supervisory tasks. People are not particularly good at monitoring, and therefore, keeping the controller active is an advantage. The system is also more flexible when the aid is used as a tool and not as a replacement, in that special cases can be more easily handled.

A decision aid can merely identify the existence of a problem or it can also give advice to help in solving the problem. Woods and Roth [1988] define good advice as more than recommending a solution. Advice needs to be given in the situations which call for it, and only when needed. If failures of attention are a problem in performing a task, an aid which focuses the operator's attention on the relevant information is an appropriate goal for the designer [Woods 1986].

The issues of usability and understandability need to be addressed. Information needs to be presented in clear formats that are easy for people to use and understand. After all, what good is providing the user with a decision aid if the user can not use it properly or does not understand what the aid is telling him? To guard against this problem, Norman [1988] advocates doing "user-centered design." He suggests the following principles of good design:

- **Make things visible:** The user should be able to tell what is going on by merely looking at the interface.
- **Provide the user with a good conceptual model:** The designer should provide the user with a mental model that is consistent and coherent.
- **Provide the user with good mappings:** The user should be able to determine the relationships between actions and their results.
- **Provide the user with feedback:** The user should receive full and continuous feedback about the results of all actions.

Another key issue is that of user acceptability. What good is a decision aid if the user does not want to use it or does not have faith in it? If the user is expected to override the computer recommendations, he should have the actual authority to do so. If the aid is used continually, care needs to be taken so that boredom is not aggravated in users when the work load is light, by reducing their workload even more [Hopkin 1988]. Often dangerous accidents and failures of attention occur when the user is less active, rather than under high workload situations.

The type of information which is revealed to the operator is another consideration of this type of aiding. If the operator receives just a warning indicating a problem, but is not told what the problem is, this is not very useful and could be found to be more of a nuisance than an aid. Also important is how the operator is alerted to a warning. For example, if he hears an annoying buzz or beep every time there is a possible conflict, and if this were to happen constantly, the buzz or beep could prove more distracting and a nuisance, and may in fact no longer serve its function if the user decides to tune it out (or even to disable it) [Norman 1988].

Also important to consider is whether the aid is working at the correct level of abstraction or whether it is supporting the correct mode of problem solving. Vicente and Rasmussen [1990a, 1990b] propose a method for interface design they call ecological interface design (EID). EID is based upon Rasmussen's skills, rules, and knowledge (SRK) framework for human performance and on his means-ends abstraction hierarchy which illustrates the functional properties of a system [Rasmussen 1986]. The SRK framework proposed that information can be detected in three ways—as signs, signals, and symbols. Information is perceived as signals when the operator detects the time-space behavior of the data. Signs are interpreted when the perceptual characteristics of the data are detected. Finally, symbols are perceived which represent concepts and have meaning. SRK claims that the way in which information is detected is related to the way it is processed [Vicente 1988]. Therefore, the three different forms of information—signals, signs, and symbols—refer to three different processing modes—skill-based, rule-based, and knowledge-based behavior. "The basic implication is that one should design interfaces in such a way as not to force cognitive control to a higher level than the demands of the task require, while at the same time providing the appropriate support for all three levels [Vicente and Rasmussen 1990a]." By providing a sentential aiding system, skill-based performance is not supported. Signs and signals are not conveyed, only symbols—the words.

The abstraction hierarchy is a tool used to describe the functional properties of a system. "Such a hierarchy describes bottom-up what components and functions can be used for, how they may serve higher level purposes, and, top-down, how purposes can be implemented by functions and components [Rasmussen 1986]." The levels in the abstraction hierarchy include physical form, physical function, generalized function, abstract function, and system purpose. The hierarchy provides a way to structure the properties to be represented in the interface, while the constraints revealed within the hierarchy enable the operator to focus his attention on the most appropriate system component by crossing through various levels of the hierarchy. In terms of the level of abstraction, different levels can be supported by the intelligent decision aid, depending on exactly which information is presented to the user. However, in order to support all levels in the

abstraction hierarchy, it would probably be necessary to supply the operator with so much verbal information that it would require too much time for him to use it properly and it would require more space than desired to implement.

Another concern is what happens if the aid is inoperational and the operator needs to function on his own. Will he lose the skill he needs to detect possible problems on his own after using an aid for a while [Woods 1986]? If the user is depending on the aid to alert him to dangers, will he stop using particular cues which he would need to perform this replaced function, but which would also help him in determining a resolution to the problem?

Therefore, we see that the following possible error modes may be introduced into a system through the use of intelligent decision-aiding:

- increased boredom of the operators leading to worsened performance;
- lack of trust in the aid;
- lack of responsibility taken by the user for override of the decision aid;
- skill reduction in the operator;
- failure of the operator to attend to important situational cues;
- lack of user acceptance of the aid;
- inability of the operator to identify problem resolutions (when they are not provided by the aid);
- worsened system performance due to the aid being a nuisance.

The next section considers the issues surrounding enhanced displays. As will be seen, many of the issues are similar to those discussed in this section. However, due to the inherent nature of these two technologies, they each have a set of issues which characterize them with respect to system domain and implementation.

Characteristic Issues of Display Enhancements

Display enhancements have been more recently proposed as an alternative to intelligent decision aiding for certain human-machine systems. Hammond, Hamm, Grassia, and Pearson [1987] and Vicente and Rasmussen [1990a, 1990b] have looked at the effects which interface design can have on modes of processing and at how interface design can be improved so as to take better advantage of natural human abilities, two of the issues involved in determining how enhanced displays can work as a tool for decision aiding.

Because of the inherent nature of enhanced displays the role of this technology in the system is that of an instrument. Again, the human operator is left in charge while the enhanced displays are designed to aid him in focusing on the correct information when it is needed. However, this is the area where the designers need to be the most careful. The question is whether, when actually implemented, the enhancements actually help the user to find, integrate, or interpret the right data at the right time. If implemented incorrectly, the enhancements can create confusion and become a hindrance. Larkin and Simon [1987] in studying pictorial representations, have found that although the following characteristics are not sufficient for a diagram, or in this case an interface display, to be useful, they are necessary for the construction of a useful pictorial representation: information to be used together should be grouped together in order to reduce the search required to find the necessary elements; location should be used to group information about particular elements to eliminate the need to match symbolic labels; and perceptual inferences should be supported. Along with these guidelines, the suggestions made by Norman [1988] for effective interface design should also be followed. Related to this issue is again the issues of usability and understandability, as discussed in the previous section.

Another concern is whether the information displayed through the enhancements actually does aid the operator in making a decision. When a conflict or potentially dangerous situation arises, do the displays merely inform the user of this possibility, or do they also help him in deciding how to resolve the problem situation? Since Woods and Roth [1988] have defined good

advice as that which is given in situations which call for it, and *only when needed*, will enhanced displays create more confusion by conveying extra information, or "advice," continually?

Woods [1986] also recommends for goal-directed knowledge representation, "possible actions are organized around the goals that they can effect, and data about pre-conditions, post-conditions (effects), constraints (side effects or inter-goal couplings) and alternative means are captured." In other words, entire contexts are considered. However, in a complex, dynamic system, the current "context" is always changing. It is possible that the display enhanced interface will not be able to isolate contexts for each task, and in doing so does not focus the operator's attention on the important information. However, Woods [1986] also claims that "if available data are organized and displayed so that the user can directly see the state of task-meaningful objects [as is the case in general for display enhanced interfaces], then natural mechanisms for focusing in on the relevant data for the current context will be more effective."

In terms of user acceptability, because of the inherent nature of enhanced displays they are different from intelligent decision aids in how the user views them as part of the system. To the user, the enhanced displays are merely a part of the user interface of the system he is working with. The display is not a computer trying to tell the person what to do (although indirectly they are). In this sense, user acceptability is viewed differently in considering enhanced displays than when considering traditional decision aids. If the user is content to work with the system interface as presented to him, the designer needs not consider what would happen if the user decides to ignore direct suggestions for actions or warning of potential crises—instead the user sees himself as determining these for himself.

Similar to intelligent decision-aiding is the question of what happens if the aid (in this case the system with the enhanced displays) is inoperational and the operator needs to function on his own. Will the user lose the skill he needs to detect possible problems on his own after using the aid for a while? Or will the use of the aid have effectively trained him to know where to look on the system display by himself? This issue is important not only in the case of an emergency or unexpected situation, but it brings up many points related to the training of operators, as well.

Finally, another issue to consider is whether the display enhancements are at the wrong level of abstraction or support the wrong mode of problem solving. The abstraction hierarchy and the modes of problem-solving associated with the SRK framework were described in the previous section. Enhanced displays have the inherent capability of better supporting these three processing modes than the decision-aiding system alternative when they are designed properly. Also, in terms of levels of abstraction, the enhanced display version of a decision aid better supports such a structured view of the system properties to be represented in the system interface by providing support for answering the questions, WHY (going up in the hierarchy) and HOW (going down). The interface for the decision aiding system does not directly support these questions or this structure.

Therefore, from this discussion we can see that the following possible error modes may be introduced into a system by the introduction of enhanced displays:

- skill reduction in the operator;
- failure of the operator to attend to important situational cues;
- inability of the operator to identify problem resolutions;
- worsened system performance due to the aid creating more confusion and being a hindrance.

Now that we have examined the issues characteristic of each type of decision-aiding technology, it is important to investigate the differences in decision-making processes found in varying levels of expertise in operators. These differences may help to identify what is needed of a decision aid intended to help a particular type of operator. They may also lend advice when determining how to best aid the training of operators.

Differences in Expert and Novice Decision-Making

Psychologists have been interested in studying the distinctions between expert and novice behavior in many different task domains. For example, Chi, et. al. [1981] have studied the differences in solving physics problems, Staszewski [1988] has looked at expertise in mental calculation (specifically, "lightning mental calculators"), and Soloway, et. al. [1988] have examined how expert and novices write (and read) computer programs. Tasks like these are concerned exclusively with the acquisition of cognitive skills. This work contrasts with that which focuses on skill acquisition of perceptual skills. It is the latter which will be examined in this section because the the research proposed here will involve human-environment interaction.

Investigating how people interact with their environment, Dreyfus, Dreyfus, and Athanasiou [1986] have witnessed five common stages of skill acquisition in the progression from novice to expert. Their research has dealt with such areas as piloting airplanes, playing chess, driving automobiles, and learning a second language. The five stages they found are novice, advanced beginner, competent, proficient, and expert.

The authors theorize that this progression from novice to expert is characterized by the following behavior. During the novice stage context-free elements and rules are learned. These elements and rules are considered context-free because they are clearly defined and easily recognized independent of the specific situation in which they occur or apply. Eventually, with practice and increased experience, novices learn to recognize "situational" elements which are context-sensitive and not objectively definable. At this point they become advanced beginners. More experience leads to the ability to adopt a plan to organize a specific situation. In this way competent behavior allows someone to improve his performance and make decisions in a hierarchical manner. Proficient behavior is characterized by the ability to "intuitively" make decisions by relating present situations to previously experienced situations. In this way the person can use expectations and previously used plans to solve new problems. Finally, an expert makes

decisions and solves problems easily and without effort. He performs naturally and does "what normally works."

This model has not yet been formalized (computationally) and has not yet been rigorously evaluated. However, it does provide a way of looking at the progression from novice to expert which considers the environmental factors and which appears to occur across a wide variety of domains.

Focusing on the final stage of acquisition, Klein [1989] has proposed a model to explain expert decision-making which consists of four main steps: 1. recognizing cases as typical; 2. understanding the situation; 3. evaluating alternatives; and 4. progressive deepening. Decision makers rely on their previous experience to recognize and classify situations as typical. Understanding a situation is comprised of recognizing four types of information—plausible goals, critical cues and causal factors, expectancies, and typical actions. Once classified, the decision maker can recall the typical way of handling that type of situation. He would use available time to evaluate whether or not an option would be appropriate. This might be done using imagery, where the decision makers would imagine implementing the option, in order to determine if anything might go wrong. If problems do arise in this imaging, the plan could be modified or even rejected. If a satisfactory plan is found, it is implemented. If a plan is rejected, a new one is selected and evaluated.

While the cognitive model here consists of familiar situations, typical actions, goals, expectancies, progressive deepening (imagining how an option will be carried out within a specific situational context), evaluation, and selection, the environmental model consists of situational cues, actions, decision points, and outcomes. Therefore, the model is sensitive to the critical cues which the decision-maker has learned to recognize. The expert decision maker is the one who has learned to distinguish which of the available cues are the critical ones. Although this work is supported by studies Klein has done in perceptually rich domains involving fireground commanders, tank platoon leaders, and design engineers, it has not been rigorously tested or completely formalized either.

Lesgold, et. al. [1988] have studied varying levels of performance in another perceptually rich domain, that of X-ray diagnosis. They have found that experts first seem to develop mental representations which in turn direct perception. Experts also appear to evoke an appropriate behavioral schema rather quickly. They know where to look and what cues to look for. In other words, they "have the ability to discriminate between relevant information and 'noise' in a given domain of action, by invoking both precepts and practice ... [Suchman 1987]." In contrast to novices, they were able to distinguish subtle differences and were more flexible in considering other possibilities based on feature detection. Lesgold, et. al. also found that performance was a nonmonotonic function of experience (similar to results found in language learning [Hetherington and Parke 1986]), that is, performance does not increasingly improve as people gain more experience. In between the stages of novice and expert, people reach a stage where their performance degrades slightly. Lesgold, et. al. studied aspects of both cognitive and perceptual learning. They proposed that the development of expertise first comes through a "perceptual tuning" in which the stimulus pattern was classified with the diagnostic decision which had the highest probability of occurring. The result of this perceptual processing would then be used for cognitive processing to resolve ambiguity. This cognitive processing can not evolve until perceptual processing has been tuned and, therefore, they propose that the development of expertise is a shift from purely perceptual decision-making to progressively deeper cognitive decision-making.

Using a different approach, DeGroot [1965] was interested in discovering the differences between players of varying degrees of expertise in the domain of chess. Specifically, DeGroot looked at chess players at the expert and Grand Master levels. Each subject was given the identical chess position and then verbal protocols were taken as he decided what move to make. Results showed that there was essentially no difference in the thought processes between the two groups—search patterns, number of moves considered, etc. The only real difference between the levels was in the quality of the move finally chosen.

Similarly, Chase and Simon [1973] conducted a study in which they replicated some of DeGroot's results and tried to isolate and define the structures (i.e., chunks) into which the information perceived by chess players is organized and stored in memory. They used three levels of players—novice, intermediate, and master. Two tasks were used in this experiment—the perception task and the memory task. The perception task required chess players to reconstruct a chess position while the original remained in view. The memory task required players to reconstruct a position from memory after being exposed to the target board for a short time (5-10 seconds). In the perception task, evidence was found indicating more rapid encoding of information for the more experienced players. In the memory task Chase and Simon found the number of pieces per chunk varied among the skill levels and that the pieces within a chunk seemed to have relationships of defence or attack, to be close together, and to be of the same color and type. Finally, an interesting result found was that when both groups of players (experts and novices) were confronted by random positions (not found in actual games), they both did equally poorly.

In contrast to most of the research described here, very little work has been done in domains involving human-environment interaction which are non-adversarial and deal mostly with skill (as opposed to cognitive processes). An example of this type of research is that done by Deakin and Allard [1991] which investigates the ability of expert and novice figure skaters in recalling elements of a figure skating routine. One important finding was evidence that figure skaters do not simply memorize the sequence of elements for a routine in the same manner that they would memorize a verbal list. Also, "expert skaters seem to have faster access to semantic memory for skating elements than do novices."

Allard has also looked at expertise in sports requiring more open skills (occurring in a moving and changing environment such as volleyball [Allard and Starkes 1980] and basketball [Allard, Graham and Paarsalu 1980]). She has found that there are many similarities in chunking and categorizing performance (thought to show the expert's ability to classify elements according to the significance to the situation) between experts in these types of sport domains and those in

more cognitive domains such as chess [Allard and Burnett 1985]. Evidence was also found that expertise in this type of domain can be split into two components—declarative knowledge (for cognitive tasks) and procedural knowledge (for playing the game).

Based upon much of this research (especially that done by Dreyfus, et. al. [1986], Klein [1989], and Lesgold, et. al. [1988]) it appears as if novices, who work more with context-free elements and rules and are not as able to identify subtle differences, are working more on a level which can be best described using rules. If this is the case, an intelligent decision aid may make more sense to them because it appears to be making decisions in a way more similar to the processes that the novices themselves use. In contrast, experts behave more intuitively and are very context dependent. Therefore, enhanced displays seem to operate in a way more consistent with how they view the domain. It will be interesting to determine through the experiment proposed in the next section whether this is indeed the case. Also, it will be interesting to see if the different types of aiding have different effects evident in training novices to become experts.

Proposed Research

The research proposed in this section will compare the two decision-aiding techniques of display enhancement and intelligent decision-aiding in the complex, dynamic domain, Star Cruiser (see Kirlik [1990] for a complete description of this task). In particular, we will study the effects that these two aiding systems have in creating possible errors modes, or performance problems, as well as any differences in performance associated with different skill levels of operators.

Decision Aid Descriptions

Although the actual modified versions of the Star Cruiser task have not yet been created, this section is intended to provide the reader with an idea of what the actual changes may involve.

Two modified versions are needed—an intelligent decision-aiding system (DA) and a display enhanced system (DE).

The DA version will provide a traditional sentential type of advice window which will display to the user a ranked list of recommended actions. The recommendations could be ranked, according to how many points could be scored if the recommendation is followed (with more points being better). Also, if the operator tries to execute an action which is undesirable (i.e., deploy a manned ship to a planet which does not support life, deploy a ship to a planet without the appropriate data or resources, load too many data or resources onto the star cruiser, etc.) the system will respond with a dialogue box pointing out the error to the operator. In contrast, the type of enhancements that might be included in the DE version include displaying an ellipse around the star cruiser designating the regions it can travel to and make if back to its star base to refuel, displaying to the operator only those types of ships that can be deployed in the solar system in which the star cruiser is orbiting, highlighting only those craft containing data or resources that can fit onto the star cruiser craft, and highlighting only those planets within orbit that a chosen craft can be deployed to.

Basis for Comparisons

Performance on the two systems can be compared by examining the points scored, the number of sessions which terminated early (due to running out of fuel, trying to load too many data or resources onto the star cruiser, or crashing into suns), the number of times craft were deployed to planets without resources or data, and the number of times manned craft were deployed to planets without life support (and unmanned craft deployed to life-sustaining planets).

Experimental Design

For the main experiment, that of comparing version DA with version DE, nine main subject groups will be required [see Figure 1]. Additionally, different levels of expertise (novice and expert) will

also be examined in regard to these nine main subject groups. Appendix A contains the hypotheses that may be tested in regards to the performance of the nine subject groups.

Anticipated Problem Areas

Extra care will need to be taken to try to ensure that neither of the two systems (DA or DE) introduces more information than the other. Also, in order to extract more general conclusions from the results found in this experiment, the Star Cruiser interface will need to be tested to determine how sensitive it is to different implementations.

		system trained on		
		unaided	DE	DA
system tested on	unaided	A	D	G
	DE	B	E	H
	DA	C	F	I

Figure 1: Subject groups for the main experiment

Preliminary Analyses

Before the modified versions of the Star Cruiser task described in the previous section can be created, a model of the Star Cruiser environment needs to be developed, which is independent of the actual information contained in the implementations of the interface. An abstraction

hierarchy is a means of doing this, without specifying strategies that should be used by the operator. A preliminary hierarchy is shown in Figure 2 (Note: the second level is still in development). Figure 3 shows the further decomposition of the **collection** function.

Also necessary before developing the decision-aiding versions of Star Cruiser, is the identification of the different types of actions available within the Star Cruiser task with which the user may need help in deciding what to do. A preliminary listing of these actions is found in Appendix B. After these actions have been identified, those actions which can be aided both perceptually (through enhanced displays) and cognitively (through sentential advice) need to be discovered, as well as the means for implementing the aiding devices for these actions.

Finally, the chosen implementations for the decision-aiding techniques will be coded in order to create the two modified Star Cruiser versions to be used in the comparison study described previously.

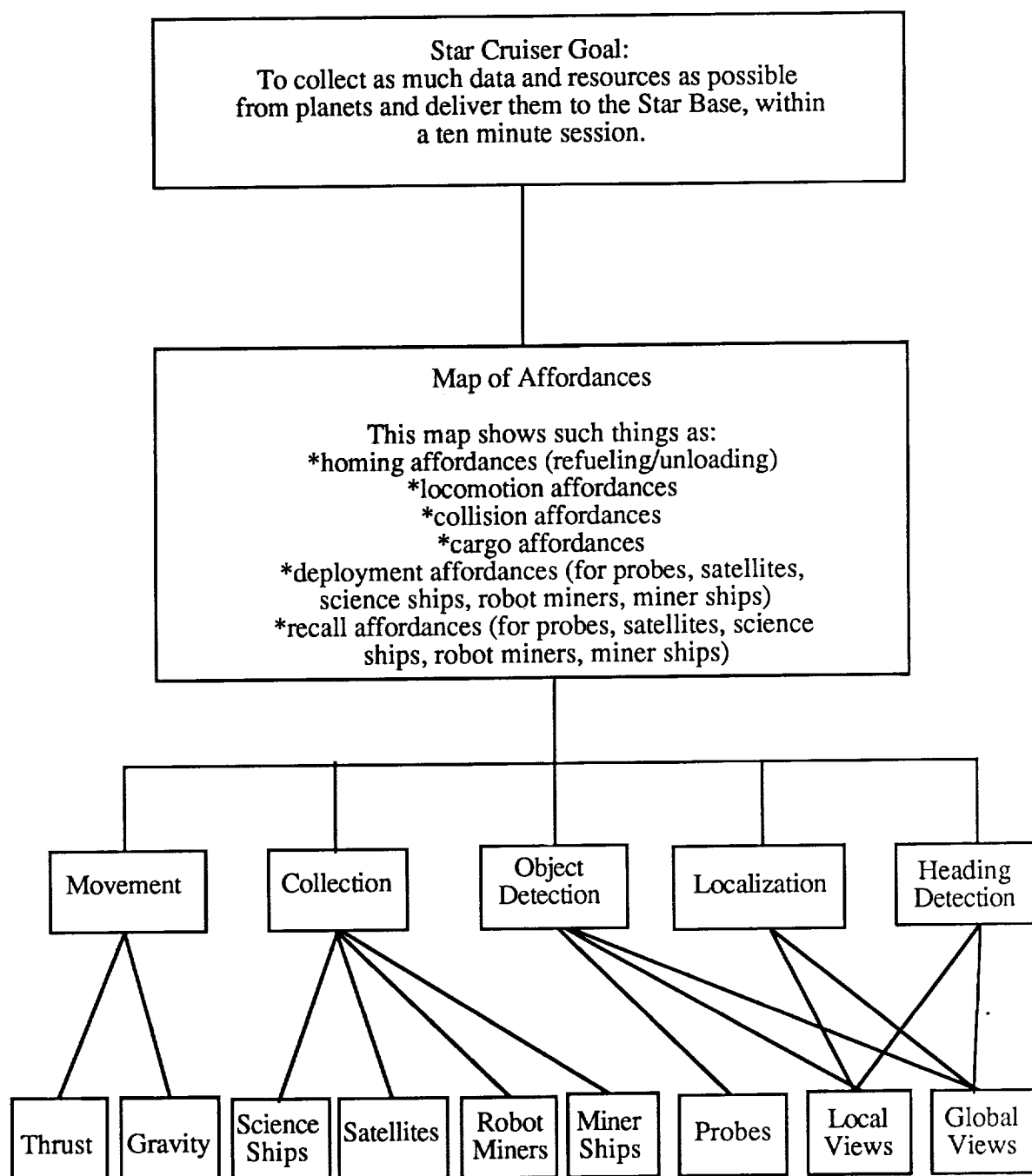


Figure 2 Preliminary Abstraction Hierarchy for Star Cruiser

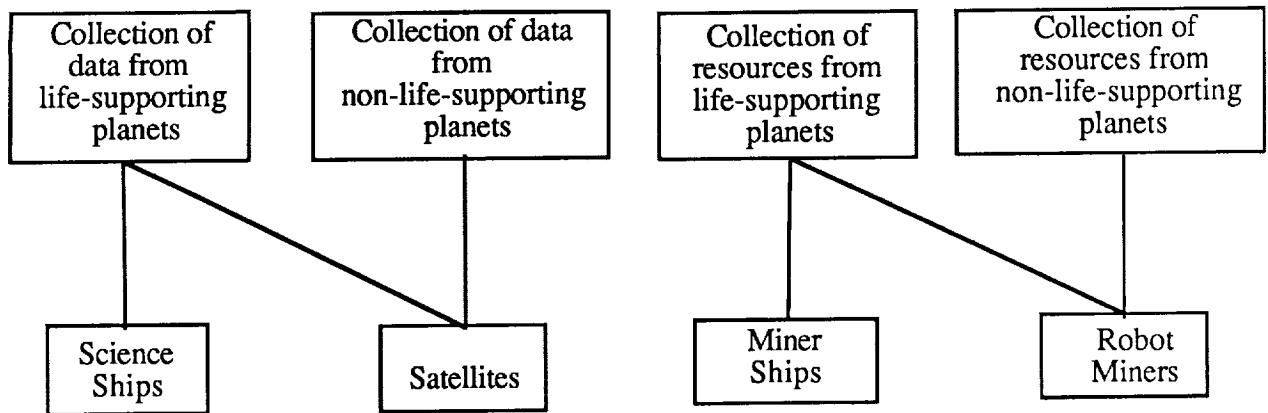


Figure 3 Decomposition of the Collection Function

Expected Contributions

The research proposed here hopes to integrate the independent research done on each of the two main decision-aiding technologies—intelligent decision-aiding and enhanced displays. It also hopes to determine which types of systems are better aided by which type of aid by carefully studying how domain characteristics interact with the characteristics and possible error modes associated with each of the two technologies. Accomplishing this task would greatly contribute to the areas of interface design and human-computer interaction, by providing guidelines for system design.

Additionally, by studying the interaction effects between operator skill level and type of decision aid, we hope to better understand the differences between expert and novice decision-making by discovering how each is better aided and under which conditions each is better aided. Finally, by studying the interaction effects between type of system trained on and type of decision aid, we hope to discover important guidelines and issues involved in the training of operators.

References

- Allard, F. and N. Burnett. 1985. Skill in sport, in *Canadian Journal of Psychology*, 39(2).
- Allard, F., S. Graham, and M.E. Paarsalu. 1980. Perception in sport: basketball, in *Journal of Sport Psychology*, 2.
- Allard, F. and J.L. Starkes. 1980. Perception in sport: volleyball, in *Journal of Sport Psychology*, 2.
- Chase, W.G. and H. A. Simon. 1973. Perception in chess, in *Cognitive Psychology*, 4.
- Chi, M.T.H., P.J. Feltovich, and R. Glaser. 1981. Categorization and representation of physics problems by experts and novices, in *Cognitive Science*, 5.
- Deakin, J.M. and F. Allard. 1991. Skilled memory in expert figure skaters, in *Memory and Cognition*, 19.
- DeGroot, A.D. 1965. *Thought and Choice in Chess*, Mouton & Co.: The Hague.
- Dreyfus, H.L., S.E. Dreyfus, and T. Athanasiou. 1986. Five steps from novice to expert, in *Mind over Machine*. The Free Press: New York.
- Hammond, K.R., R.M. Hamm, J. Grassia, and T. Pearson. 1987. Direct comparison of the efficacy of intuitive and analytical cognition in expert judgement, in *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-17, No. 5.

Hetherington, E.M. and R.D. Parke. 1986. *Child Psychology*, McGraw-Hill: New York.

Hopkin, V.D. 1988. Air traffic control, in *Human Factors in Aviation*. Academic Press: New York.

Kirlik, A. 1990. *Star Cruiser*, a design specification document submitted to Sterling Software.

Klein, G.A. 1989. Recognition-primed decisions, in *Advances in Man-Machine Systems Research, Vol. 5*, W.B. Rouse (ed.). JAI Press: Greenwich, CT.

Larkin, J.H. and H.A. Simon. 1987. Why a diagram is (sometimes) worth ten thousand words, in *Cognitive Science*, 11.

Lesgold, A., H. Robinson, P. Feltovich, R. Glaser, D. Klopfer, and Y. Wang. 1988. Expertise in a complex skill: diagnosing X-ray pictures, in *The Nature of Expertise*, Chi, Glaser, and Farr (eds.). Lawrence Erlbaum Associates: Hillsdale, NJ.

Norman, D.A. 1988. *The Psychology of Everyday Things*. Basic Books, Inc.: New York.

Rasmussen, J. 1986. *Information Processing and Human-Machine Interaction*. North-Holland: New York.

Soloway, E., B. Adelson, K. Ehrlich. 1988. Knowledge and processes in the comprehension of computer programs, in *The Nature of Expertise*, M.T.H. Chi, R. Glaser, and M.J. Farr (eds.). Lawrence Erlbaum Associates: Hillsdale, NJ.

- Staszewski, J.J. 1988. Skilled memory and expert mental calculation, in *The Nature of Expertise*, M.T.H. Chi, R. Glaser, and M.J. Farr (eds.). Lawrence Erlbaum Associates: Hillsdale, NJ.
- Suchman, L.A. 1987. *Plans and Situated Actions*, Cambridge University Press: Cambridge, MA.
- Vicente, K.J. 1988. Adapting the memory recall paradigm to evaluate interfaces, in *Acta Psychologica*, 69.
- Vicente, K.J. and J. Rasmussen. 1990a. Ecological interface design: theoretical foundations, unpublished manuscript.
- Vicente, K.J. and J. Rasmussen. 1990b. The ecology of human-machine systems II: mediating "direct perception" in complex work domains, in *Ecological Psychology*, 2(3).
- Woods, D.D. 1986. Paradigms for intelligent decision support, in *Intelligent Decision Support in Process Environments*, E. Hollnagel, et. al. (eds.). Springer-Verlag: Berlin.
- Woods, D.D. and E.M. Roth. 1988. Cognitive systems engineering, in *Handbook of Human-Computer Interaction*, M. Helander (ed.). Elsevier Science Publishers

Appendix A

Hypotheses for the Star Cruiser Experiments

(Each letter here represents the performance of the corresponding subject group found in Figure 1.
DE = display enhanced version. DA = intelligent decision-aiding version)

- A<B: equal training (unaided), DE leads to better performance than original unaided
- A<C: equal training (unaided), DA leads to better performance than original unaided
- A<D: something learned during training (on DE) is good and transferred (both tested on unaided)
- A<E: something about DE is better than unaided
- A<F: either something about training w/DE or testing on DA leads to better performance than unaided
- A<G: something learned during training (on DA) is good and transferred (both tested on unaided)
- A<H: either something about training w/DA or testing on DE leads to better performance than unaided
- A<I: something about DA is better than unaided
- B>C: same training (unaided), DE leads to better performance than DA
- B>D: DE leads to better performance (nothing learned or not enough to offset during training)
- B<D: more transferred during training w/DE
- B<E: (want a little bit) but close to equal performance, diff. comes from training (equal testing—DE)
- B F*
- B>G: DA leads to better performance (not enough transferred during training to offset)
- B<G: more transferred during training w/DA
- B<H: something from DA transferred during training to improve performance on DE (better than unaided)
- B>I: DE leads to better performance than DA w/o training on it
- C<D: something during training w/DE transferred and led to better performance
- C>E: DA leads to better performance than DE even w/o training
- C<F: something during training w/DE leads to better performance on DA than training unaided
- C>G: DA leads to better performance (not enough transferred during training)
- C<G: more transferred during training w/DA
- C H*
- C<I: want a little better—shows better performance due to training w/DA (equal testing—DA)

- D<E: otherwise DE during training transferred and accommodated (equal training—DE)
 D<F: DA leads to better performance than unaided w/ equal training on DE
 D>G: something during training w/DE transferred and improved performance on unaided more than training w/DA

D H

- D<I: otherwise more transferred during training w/DE
 E>F: otherwise equal training (on DE) but DA improves performance
 E>G: otherwise more transferred during training w/DA than training and testing on DE
 E>H: otherwise something during training w/DA transferred and led to improved performance on DE
 E>I: something about DE better than DA

F G

F H

- F<I: otherwise something about DE during training transferred and led to improved performance over DA
 G<H: equal training (DA), DE better performance
 G<I: equal training (DA), DA better performance than unaided
 H<I: otherwise equal training (DA), DE better performance than DA

Appendix B

Listing of Star Cruiser Actions

The following is a listing of possible actions available on the Star Cruiser application. A total of five factors are discussed for each. The first paragraph under each action heading explains when the action **can** and **cannot** be performed or, in other words, when the program will allow the user to perform the action and when the user will not be allowed to do so. The second paragraph explains when the user **should** and **should not** perform the actions. This section is based on my own experience with the application and it details those times performing an action can be beneficial or when it can be detrimental to the user's performance. The third paragraph under each heading details what **perceptual support** exists for that action. It explains what support currently exists, whether it is satisfactory or not, and possible improvements. Once again, this is based on my experience with Star Cruiser and thus may differ from someone else's opinion.

There are several characteristics about Star Cruiser that one should keep in mind as s/he read through this listing. The first is that whenever one of Star Cruiser's tools needs to be deployed or recalled, the user must be viewing the map corresponding to Star Cruiser's location (i.e., in galaxy - global map; in solar system - local map). In addition, "movement" of Star Cruiser can only occur if it is not docked at Star Base or in an orbit. If it is, then Star Cruiser must first be taken out of orbit or pulled away from Star Base before it can travel freely.

Finally, it should be realized that many of Star Cruiser's movements are in preparation for the user to perform some other action. Most movements are the result of the user wanting either to deploy or recall one of Star Cruiser's tools or have Star Cruiser return to Star Base in order to refuel/unload cargo. Also, the choice (direction, speed) of movements may also depend on what information is obtained through viewing the global or local maps. It becomes apparent that, due to these factors, Star Cruiser's movement through the galaxy and solar systems is usually quite dependent on other actions that the user has just performed or wishes to do in the near future.

Deploy Probe

A user may deploy a probe anytime except when docked at the Star Base. In other words, the only time a probe can not be deployed is when the Star Cruiser is docked at the Star Base.

Deploying probes has no effect on points or fuel consumption and therefore can be done almost anytime the user wants to. It is advisable to deploy probes when the user has difficulty locating the 9th orbital in order for the Star Cruiser to orbit a sun. Deploying a probe is also useful when the user wishes to know the amount of data/resources in a particular solar system before visiting it. If the user has little difficulty in obtaining orbit with the Star Cruiser and/or doesn't need to know the amount of data/resources before visiting a particular solar system, then probe deployment has little value.

There is currently no perceptual support that informs the user when it is possible or best to deploy a probe. Since the deployment of a probe has no effect on fuel consumption or the user's score, there is no real need to inform the user when a probe should be or shouldn't be deployed. Blacking out the probes from the selection bar at the top of the screen can be of useful in preventing the user from trying to deploy a probe while docked at the Star Base.

Recall Probe

A probe in orbit around a sun may only be recalled by the user if the Star Cruiser is in orbit around that same sun and the user is viewing the system in local mode. If the user is viewing the galaxy, if the Star Cruiser is not in orbit in the same solar system as the probe, or if there are no probes in the solar system currently being visited by the Star Cruiser, then no probes can be recalled.

Probes only need to be recalled if they are to be used somewhere else in the galaxy. If that is the case, then it is suggested that the user recall a probe when all data/resources have been collected from the particular solar system and/or the user no longer needs assistance in identifying the 9th orbital. If the user still has trouble getting the Star Cruiser in orbit, then it is recommended that the user do not recall the probe.

If the program determines that the user would like to deploy another probe, but none are available, then it can highlight a probe that has already been deployed and is present in a system that contains no more data/resources. This would let the user know immediately the most ideal probes to recall. This, though, would be rather redundant in that the user can simply view the pie-chart present on the suns in global mode to determine which solar systems no longer contain any data/resources. Therefore, a significant amount of perceptual support already exists in helping the user to identify the probes which can or need to be recalled.

Deploy Satellite/Robot Miner

A Satellite/Robot Miner can be deployed only when the Star Cruiser is in orbit around a sun which contains planets and the user is currently viewing that particular solar system's local map. If the Star Cruiser is not in orbit or in a solar system or if that solar system doesn't contain any planets or if the user is not viewing the local map of the solar system containing the Star Cruiser, then deployment of a Satellite/Robot Miner is not possible.

A Satellite/Robot Miner should be deployed to any blue planets containing data/resources that need to be collected. They should also be deployed to any green planets that contain data/resources in order to prevent any Science Ships/Minerships from automatically collecting too much data/resources and thus preventing themselves from being recalled for fear of overloading the Star Cruiser with too much data/resources. The only time it isn't beneficial to deploy Satellites/Robot Miners is when there are no planets in the solar system which contain any data/resources.

No extensive perceptual support exists that helps the user decide whether or not to deploy a Satellite/Robot Miner and if so, which planet to deploy it to. The only clues that are present to the user are the Star Cruiser's gauges that relate, qualitatively, how much data/resources is currently on board the Star Cruiser. Because this action is one of the more important ones performed by the user, better support should be present. One possibility is to automatically highlight a Satellite/Robot Monitor (accompanied with an auditory signal) to signal to the user that one can be deployed. In addition, by highlighting a particular planet, the user would also know where best to deploy the Satellite/Robot Miner. It is questionable whether or not planets that contain too much data/resources for Star Cruiser to handle should be highlighted. The absence of any highlighted Satellites/Robot Miners would indicate that the user should not deploy any.

Recall Satellite/Robot Miner

When Star Cruiser is in orbit in a solar system where Satellites/Robot Miners are deployed to planets and the user is viewing the solar system in local mode, then those deployed Satellites/Robot Miners may be recalled. If the Star Cruiser is not in orbit in a solar system where Satellites/Robot

Miners have been deployed or if it is the galaxy or if the user is viewing the galaxy, then Satellites/Robot Miners may not be recalled.

Satellites/Robot Miners should be recalled when they are finished collecting the data/resources from their particular planets. Care must be taken not to recall them if they have collected so much data/resources that it would overload the Star Cruiser. Therefore, the user should recall the Satellites/Robot Miners before they complete their missions if, in not doing so, they run this risk.

As with deployment, the only support present to the user in making the decision when to recall a Satellite/Robot Miner are the Star Cruiser's gauges. Once again, the need is present for better perceptual support. Deployed Satellites/Robot Miners can be highlighted when the program feels it is best to recall them. This, of course, will depend on the amount of data/resources currently present aboard Star Cruiser and how much the Satellites/Robot Miners have and/or can collect at their planets. If they can collect all of the data/resources without resulting in an overload when recalled, then the program can highlight the Satellites/Robot Miners when they have completed their missions. If the planets contain too much data/resources, then the program can highlight them before they finish, thus informing the user that they need to be recalled as soon as possible. If the Satellites/Robot Miners have already collected too much for one reason or another, then the program will not highlight them until the Star Cruiser has unloaded it's current haul at Star Base and returned to the current solar system.

Deploy Science Ship/Minership

Science Ships/Minerships can be deployed under the same circumstances as Satellites/Robot Miners with the one exception that green planets must be present in the solar system since they may only be deployed to a planet which "supports life." If no green planets are present in the solar system, or if any of the other conditions similar to Satellites/Robot Miners are not met, then Science Ships/Minerships cannot be deployed.

Science Ships/Minerships should be deployed whenever green planets are present in the solar system and contain data/resources. They should, however, not be deployed if the total data/resources that will be collected by any one Science Ship/Minership will overload the Star Cruiser. Therefore, if this risk exists, then Science Ships/Minerships should not be deployed.

*** Refer to the discussion of perceptual support for Satellites/Robot Miners. The issues discussed there may also be applied to the Science Ships/Minerships. ***

Recall Science Ship/Minership

*** All issues discussed under Recall Satellite/Robot Miner may also be applied here. The one exception is in regard to highlighting the Science Ships/Minerships. Since these ships will move from green planet to green planet, collecting all available data/resources and because the possibility exists that these ships may collect so much data/resources that they would even overload an empty Star Cruiser, the program should inform the user when to deploy multiple Science Ships/Minerships (by highlighting them) in order to prevent this. This prevention is accomplished by dividing the available amount of data/resources amongst various ships so that the smaller portions may still be loaded onto Star Cruiser. ***

Move Star Cruiser Into Orbit

Star Cruiser can be placed in orbit whenever it is present in a solar system. If it is moving about the galaxy, then Star Cruiser cannot be placed into orbit around any of the suns.

The user should place Star Cruiser into orbit around a sun whenever satellites, robot miners, science ships, and/or minerships are to be deployed in that particular solar system. Also, if any of them, along with probes, are to be recalled from that same system, then Star Cruiser must also be in orbit. It becomes unnecessary to place the Star Cruiser in orbit in a solar system if there are no planets present or if the need/desire to deploy or recall any ships or probes does not exist in that system.

The program already gives some hints to the user as to when it is necessary to obtain orbit with the Star Cruiser. The pie-charts (in local mode) representing the amount of data/resources available on a planet are an indication as to when ships should be deployed or recalled. These hints, however, do provide a direct mapping between the desired situation (deploy/recall ship) and the means with which to obtain the situation (put Star Cruiser in orbit). Therefore, more support is needed. One possibility is to highlight both Star Cruiser and where it should be (9th orbital). This, though, would have the drawback of showing the user exactly where the orbital is located thus making the action almost too simple to perform and also removing one of the probe's functions (identify 9th orbital). Displaying a message such as "Achieve Orbit" on the screen, which should be just as informative, would be a better option in that it would not simplify the task or remove any functions from the Star Cruiser's tools.

Move Star Cruiser Out Of Orbit

This action can only be performed if the Star Cruiser is already in orbit in some solar system and the user is currently viewing that same system.

The Star Cruiser should be moved out of orbit if the user has completed the task of either recalling as many deployments as desired or if the user wishes to exit the solar system for some reason such as moving to a new solar system or going to the Star Base. It is advisable, however, that Star Cruiser remain in orbit in order to recall as many deployments as possible as long as the risk of overloading on data/resources does not exist or there is no threat of running out of fuel.

No perceptual support exists that aids the user in determining when is the most opportune time to move out of orbit. None is really needed either. If enough support exists which informs the user of other actions to perform with Star Cruiser (i.e., dock at Star Base, recall satellite, etc.), then the user should know that in order to perform those tasks, Star Cruiser must or must not be in orbit. If the user does not know this though, a simple message can be used to provide instruction.

Move Star Cruiser Into Solar System

If Star Cruiser is moving about the galaxy, the user then has the option of moving it into a solar system. Star Cruiser cannot move directly from one solar system to another without first entering the galaxy. Nor can Star Cruiser enter a solar system if it is docked at Star Base. It must first pull away from Star Base, then it may enter a solar system.

It is beneficial to have Star Cruiser enter a solar system if that system contains any data/resources that the user wishes to collect. Thus, if the user wants to deploy any ships, then Star Cruiser must

first enter the solar system before anything else can be done. This also holds true if the user wants to recall any ships. The user should try to prevent Star Cruiser from entering any solar systems if the current task is to get Star Cruiser to Star Base so it can dock. This is critical if Star Cruiser is low on fuel since it may not be able to reach Star Base if it keeps entering and exiting solar systems.

The only support in determining when to enter a solar system that the program provides the user with are the pie-charts that may be located on the suns in the global mode. Noticing whether or not a particular solar system contains any data/resources can help the user decide if it is worth entering. There may be many solar systems, however, that contain data/resources. Therefore, the program should also suggest to the user which particular solar system Star Cruiser should enter. This can be done by printing the message "Enter Highlighted Solar System" along with highlighting a particular sun.

Move Star Cruiser Into Galaxy

If Star Cruiser is in some solar system, but is not in orbit, then the user may move Star Cruiser directly into the galaxy without having to perform any other intermediate tasks such as moving Star Cruiser out of orbit first.

When the user has completely loaded up Star Cruiser with data/resources and has already taken it out of orbit, then Star Cruiser should be moved into the galaxy so that it can make its way to Star Base. In addition, whenever Star Cruiser no longer needs to remain in a solar system, it should be moved into the galaxy so that it may travel to another system or to Star Base. Star Cruiser should more than likely not move into the galaxy if there still remains more data/resources that can be collected without causing an overload of Star Cruiser and if Star Cruiser is not at risk of running out of fuel.

There is no direct assistance provided to the user that says when Star Cruiser should exit the solar system and enter the galaxy. However, the pie-charts depicting the available data/resources shown on the planets, or their absence, should help the user determine whether or not it is worth staying in the solar system. In addition, the fuel gauge and Star Cruiser's gauges showing its remaining capacity for data/resources also help the user decide if Star Cruiser need to move into the galaxy so it can go and dock at Star Base. This, though, is generally enough support. Other assistance such as informing the user to dock at Star Base should provide further help in determining when to enter the galaxy.

Dock Star Cruiser At Star Base

The user may only dock Star Cruiser at Star Base if Star Cruiser is present in the galaxy and the user is viewing the global map. If Star Cruiser is in any solar system, then it cannot dock at Star Base.

Star Cruiser should dock at Star Base whenever it cannot carry any more data/resources or whenever it is about to run out of fuel. If Star Base is nearby, though, and Star Cruiser still can carry more data/resources without becoming overloaded and still has plenty of fuel, it is sometimes good strategy to dock at Star Base to unload the cargo and refuel. This proves beneficial when it comes time to have Star Cruiser journey to those solar systems which are far from Star Base. Star Cruiser should not be forced to dock at Star Base if it is not necessary if the base is far away. Since these missions have a time limit, actions of this nature will only waste that time.

The fuel gauge and Star Cruiser's data/resource gauges help the user determine when it is necessary to have Star Cruiser dock at Star Base. This should generally be enough support. During high workload situations, however, the user may forget to check these gauges. Therefore, as a safety precaution, it is probably wise to display some message informing the user that Star Cruiser better dock at Star Base. This would appear only under "must"-situations. The user should be allowed to determine whether or not Star Cruiser should dock without the use of any other additional information besides the gauges.

Have Star Cruiser Leave Star Base

The user can have Star Cruiser leave Star Base right after it has docked there.

Star Cruiser should be made to leave Star Base right after docking since no other actions can be performed until it has done so. The only time it would not be necessary to leave Star Base is when all data/resources have been collected, thus ending the scenario.

No perceptual support informing the user to pull Star Cruiser away from Star Base is needed. The fact that nothing else can be accomplished until Star Cruiser's departure provides enough of a forcing function to remind the user to do so.

View Galaxy (Star Cruiser In Solar System)

The galaxy may only be viewed if Star Cruiser is in orbit in some solar system. The only other time that the global map is viewed is when Star Cruiser is traveling through the galaxy itself.

This action's purpose is merely to gather information about various states of the system. Such items that may be checked by the user include the collection status of the total amount of data/resources in a different solar system; the distance from the current solar system to Star Base; or the proximity/location of other solar systems to the current system. This action is useless if the user does not desire any such information.

Because this action merely provides information to the user (it does not alter the system states in anyway), no perceptual support is required. If the user desires some piece of information that can only be gathered through viewing the global map, then the user will select that option. If it cannot be selected, then the user will realize that Star Cruiser is not in orbit and that it may be simpler to just move Star Cruiser into the galaxy. Since there is no way of determining which information the user would like to have access to, it is difficult to have the program support the decision to view the global map.

View Solar System (Star Cruiser In Galaxy)

If a probe has been deployed to a particular solar system or if Star Cruiser has previously visited it, then that system may be viewed while Star Cruiser is traveling around the galaxy. The only other method for viewing a solar system is to have Star Cruiser enter the system.

This action merely provides information to the user. Such information may include the number of planets in a particular solar system or the collection status of deployments in that system. If no information is desired about a particular system, then the user need not perform this task.

No perceptual support for this action is needed. Since no changes are being made to the system states and it is difficult to know exactly when the user desires information, let alone what kind, it would be almost pointless to try to provide any support for deciding when to perform this action.